# INTERPOLATION OF GREEN COLOR IN FRAMES FROM CMOS IMAGE SENSOR

#### Todor Djamiykov, Nikolay Nenov

Department of Electronics, Technical university, 8 Kliment Ohridski, 1756 Sofia, Bulgaria, phone: +359 2 965 32 69, e-mail: tsd@vmei.acad.bg nenov@mail.bg

#### Abstract

The paper presents an algorithm for interpolation of the green pixels in frames from CMOS image sensor. The result is a monochrome frame calculated directly from the data from the primary BAYER matrix in the sensor. An algorithm and VHDL module work in real time were developed. The aim is simple and easy for usage algorithm, which uses small amount of macro-cells, to be implemented in FPGA. The algorithm is a part of optoelectronic system for measurements of the plane coordinates of a light spot. The system is based on PLD. The monochrome frame calculated with this module is used by other image processing algorithms in the system. FPGA XC3S400 by XILINX is used.

Keywords: CMOS image sensor, VHDL, Color interpolation, Real-time applications,

#### **1. Introduction**

In optoelectronic systems a lot of image processing algorithms are used for reaching the desired result. Usually the source is CMOS image sensor. All image processing algorithms use the data from this CMOS image sensor. The light in the image sensor is received by lots of small photodiodes. There are separate photodiodes for the three main colors in the image: Red, Green and Blue. The "color" sensors are even distributed over the surface of the sensor in primary Bayer matrix. The first image processing in the optoelectronic system is color interpolation. In this way original colors in the frame are restored from the Bayer matrix. The Image processing algorithms are simpler if monochrome frames are used. Because of this second step in image processing is RGB to grayscale conversion. The proposed algorithm generates monochrome frame directly from data coming from primary Bayer matrix in the CMOS image sensor. The algorithm and firmware of the entire optoelectronic system are developed in VHDL using WebPack 6.

## 2. Standard method for RGB to Grayscale conversion

All CMOS image sensors have primary Bayer matrix. The sensor elements are read pixel by pixel and line by line. The frame outgoes the sensor in the same way - pixel by pixel and line by line. After color interpolation intensity of each pixel in the image can be calculated according to the equation 1.

(1) Y=0.59.G+0.31.R+0.11.B

This equation is complicated for implementation in FPGA, if it have to work in real time, and requires mush resources. Because of this there is necessity of simple algorithm for RGB to grayscale image conversion.

# **3. Module for Green Interpolation**

The CMOS image sensor in the optoelectronic system is OV2610 by OmniVision. It has resolution of 1600 x 1200 pixels. The green color gives main part of the information for the image to the human eyes. Because of this a great number of pixels are "green" pixels – 50% of all pixels. This fact is used in the proposed algorithm. In output frame green pixels are not changed, but the pixels between them are calculated according to equation 2.

$$(2) \qquad Gx = \frac{Gn + Gn + 1}{2}$$

Gn – green pixels in the Bayer matrix Gx – red or blue pixels in the Bayer matrix

The algorithm uses only two neighbor green pixels on the same line around Gx. In this way usage of additional video memory is avoided.

Figure (1) shows the structure of the primary Bayer matrix. The output monochrome frame is shown on Figure (2).

B	G	B	G	B	G
G	R	G	R	G	R
B	G	B	G	B	G
G	R	G	R	G	R
B	G	B	G	B	G
G	R	G	R	G	R

Figure 1: Structure of the primary Bayer matrix

Χ	GO	(G0+G1)/2	G1	(G1+G2)/2	<b>G2</b>
<b>G3</b>	(G3+G4)/2	<b>G4</b>	(G4+G5)/2	G5	X
•••	•••	•••	•••	•••	•••
Gn	(Gn+Gn+1)/2	Gn+1	(Gn+1+Gn+2)/2	Gn+2	X

Figure 2: Structure of the frame after Green interpolation

#### 4. Results

Figure 3 uses numbers to show how the algorithm works. On Figure 3a - a content of Bayer pattern image is shown. It also shows the intensity of the green pixels. Figure 3b shows the result of the algorithm. FPGA uses binary numbers and this is the reason that calculated values in the monochrome image may have deviation around the real value.

B	<b>G(2)</b>	B	<b>G(7</b> )	B	<b>G(23)</b>
<b>G(5)</b>	R	<b>G(50)</b>	R	<b>G(18)</b>	R
B	<b>G(24)</b>	B	<b>G(100)</b>	B	<b>G(80)</b>
<b>G(8)</b>	R	<b>G(32)</b>	R	<b>G(67)</b>	R
B	<b>G(58)</b>	B	<b>G(61</b> )	B	G(35)
<b>G</b> (22)	R	<b>G</b> ( <b>47</b> )	R	<b>G(90)</b>	R

Figure 3a: Content of the Bayer pattern image

X	2	4	7	15	23
5	27	50	34	18	Χ
X	24	62	100	90	80
8	20	32	<b>49</b>	67	Χ
X	58	59	61	48	35
22	34	47	68	90	X

Figure 3b: Content of the output monochrome image

A MathCAD simulation of the algorithm is shown on Figure 4. A real monochrome frame calculated with this algorithm is shown on Figure 5. This is a frame generated by the optoelectronic system and displayed on a monitor.







**4b** – output monochrome image





Figure 5: Real monochrome frame from the optoelectronic system

The resources used by the module for green interpolation is shown on Figure 6.

Device utilization summary:

\_\_\_\_\_

Selected Device: 3s400pq208-4

Number of Slices:	32 out of	3584	0%
Number of Slice Flip Flops:	41 out of	7168	0%
Number of 4 input LUTs:	41 out of	7168	0%
Number of bonded IOBs:	21 out of	141	14%
Number of GCLKs:	1 out of	8	12%

Figure 6: Used resources in FPGA

Output clock signal of the OV2610 is 20MHz. This means that the module for green interpolation must work faster than 20MHz to be correct in the calculations. Time characteristics of the module, estimated by WebPack, are shown on Figure 7.

Timing Summary:

Speed Grade: -4

-----

Minimum period: 5.152ns (Maximum Frequency: 194.099MHz) Minimum input arrival time before clock: 6.250ns Maximum output required time after clock: 6.271ns Maximum combinational path delay: 7.369ns

Figure 7: Time characteristics of the module for green interpolation

# **5.** Conclusion

The goal was simple algorithm and module for RGB to monochrome conversion to be developed. There were also requirements for small module uses few macro-cells in the FPGA. This algorithm calculates monochrome image directly from the primary Bayer image. In this way color interpolation, which is harder for computation, is avoided. The algorithm is thoroughly adapted for implementation in FPGA and work in real time.

## 6. References

1. Hamblen J.O, *Rapid Prototyping of digital systems*, Kluwer Academic Publishers, Boston / Dordrecht / London, 2001.

2. OV7620 Product specifications - REV0.9 (10/19/99) Omnivision Technologies INC.

3. Roth C. H., Digital system design using VHDL, PWS Publishing Company 1997.

4. Smith S. W., *The scientist and engineer's guide to Digital Signal Processing*, California Technical Publishing / San Diego, 1999.